

Universidad de Buenos Aires - FIUBA
66.20 Organización de Computadoras
Trabajo práctico 1: Programación MIPS
2º cuatrimestre de 2020

\$Date: 2020/10/27 23:05:14 \$

1. Objetivos

Familiarizarse con el conjunto de instrucciones MIPS y el concepto de ABI, extendiendo un programa que resuelva el problema descrito a continuación.

2. Alcance

Este trabajo práctico es de elaboración grupal, evaluación individual, y de carácter obligatorio para todos alumnos del curso.

3. Requisitos

El trabajo deberá ser entregado personalmente, en la fecha estipulada, con una carátula que contenga los datos completos de todos los integrantes, un informe impreso de acuerdo con lo que mencionaremos en la sección 6, y con una copia digital de los archivos fuente necesarios para compilar el trabajo.

4. Descripción

El programa a desarrollar deberá procesar un *stream* de texto compuesto por una cantidad arbitraria de líneas de longitud arbitraria. A medida que el programa avance en la lectura de cada línea, deberá calcular el hash del contenido de la misma, e imprimir en la salida el valor del hash seguido del contenido de la línea.

Por ejemplo, dado el siguiente flujo de entrada:

```
$ cat input.txt
66.20 Organizacion de Computadoras
TP 1 - Segundo Cuatrimestre, 2020
```

Archivo de prueba TP 1.

Al ejecutar el programa la salida será:

```
$ tp1 -i input.txt -o -
0xcc2b6c5a 66.20 Organizacion de Computadoras
0xcb5af1f1 TP 1 - Segundo Cuatrimestre, 2020
0x4c4b4f0b
0xd788c5a5 Archivo de prueba TP 1.
```

4.1. Ejemplos

Primero, usamos la opción `-h` para ver el mensaje de ayuda:

```
$ tp1 -h
Usage:
  tp1 -h
  tp1 -V
  tp1 -i in_file -o out_file
Options:
  -V, --version      Print version and quit.
  -h, --help         Print this information and quit.
  -i, --input        Specify input stream/file, "-" for stdin.
  -o, --output        Specify output stream/file, "-" for stdout.
Examples:
  tp1 < in.txt > out.txt
  cat in.txt | tp1 -i - > out.txt
```

A continuación, ejecutamos algunas pruebas: primero, veamos qué sucede cuando el archivo de entrada está vacío,

```
$ ./tp1 -o salida.txt </dev/null
$ ls -l salida.txt
-rw-r--r-- 1 leandro leandro 0 Oct 20 20:14 salida.txt
```

Aquí puede verse que el programa se comporta según lo esperado, ya que cuando la entrada está vacía, la salida lo estará también.

Veamos qué ocurre al ingresar un archivo con una única línea, la cual contiene un sólo carácter:

```
$ echo 1 | ./tp1 -o -
0x91ff4b5b 1
```

Lo mismo debería ocurrir si la entrada se encuentra alojada en el sistema de archivos:

```
$ echo 1 >entrada.txt
$ ./tp1 -i entrada.txt -o -
0x91ff4b5b 1
```

5. Implementación

El programa a desarrollar constará de una mezcla entre código MIPS32 y C, siendo la parte escrita en *assembly* la encargada de calcular el hash de un bloque de bytes pasado por parámetro. El formato de dicha función será:

```
void string_hash_more(string_hash *, char *, size_t);
```

En donde `string_hash` es un tipo de datos usado para mantener el contexto de la operación de cálculo del hash (opaco para el usuario). En el archivo `hash.c` puede encontrarse una implementación de referencia en lenguaje C de esta función de hash.

6. Informe

El informe deberá incluir:

- Documentación relevante al diseño e implementación del programa;
- Comando(s) para compilar el programa;
- Las corridas de prueba, con los comentarios pertinentes;
- El código fuente, en lenguaje C y MIPS;
- El código MIPS32 generado por el compilador¹;
- Este enunciado.

7. Regresiones

El programa deberá pasar todas las regresiones definidas en el código fuente suministrado en este TP [1]:

```
$ make
cc -Wall -g -o regressions regressions.c hash.c hash.S
:
$ make test
./regressions
0xcc2b6c5a 66.20 Organizacion de Computadoras
0xcb5af1f1 TP 1 - Segundo Cuatrimestre, 2020
0xcb5af1f1
0xd788c5a5 Archivo de prueba TP 1.
0x91ff4b5b 1
```

Asimismo deberá usarse el modo 1 del sistema operativo para manejo de acceso no alineado a memoria [2].

8. Entrega de TPs

La entrega de este trabajo deberá realizarse usando el campus virtual de la materia [3]. Asimismo, en todos los casos, estas presentaciones deberán ser realizadas durante los días martes. El *feedback* estará disponible de un martes hacia el otro, como ocurre durante la modalidad presencial de cursada.

Por otro lado, la última fecha de entrega y presentación para esta trabajo será el martes 10/11.

¹Por motivos prácticos, en la copia impresa sólo es necesario incluir la primera página del código assembly MIPS32 generado por el compilador.

Referencias

- [1] Código fuente para realizar el Trabajo Práctico.
<https://drive.google.com/file/d/1Jdz3b0AoBwipKAAsjP7qsDqR-hSSM0ik/view>
- [2] Controlling the kernel unalignment handling via debugfs,
<https://www.linux-mips.org/wiki/Alignment>.
- [3] Aula Virtual - Organización de Computadoras 86.37/66.20 - Curso 1 - Turno Martes.
<https://campus.fi.uba.ar/course/view.php?id=649>